First SDP Workshop

- **Goals**
  - Bring together leading-edge researchers and practitioners
  - Encourage brainstorming and out-of-box thinking
  - Inform the research agenda
  - Raise the visibility of SDP
  - Involve Federal agencies and research community
- **Participants from**
  - 14 universities and non-profits
  - 8 commercial enterprises
  - 11 Government agencies

Panel Organization

- **Future of Software and Software Research**

  Software's rapid penetration throughout the economy challenges fundamentals underlying software research and how research is conducted.

- **New Software Development Paradigms**

  Software is increasingly the universal integrator for large-scale and possibly network-centric distributed "systems of systems." Paradigms are needed for systems that are very complex and larger than previously built systems.

- **Software for the Real World**

  There are vast numbers of embedded systems with behavior constrained by the physical world. We need principled methods for composition and integration of such constraints with conventional functional requirements and components.

- **Software for Network-Centric Distributed Systems**

  Challenges include latency hiding, partial failure, causal ordering, dynamic service partitioning, and distributed deadlock avoidance. Techniques for end-to-end quality of service frameworks, multi-level distributed resource management, and adaptive and reflexive middleware are also needed.

2

Expanding Challenges for Software Design

- From <u>Unattended</u>   to   <u>Human-Centric</u>
- From <u>Legacy</u>   to   <u>Net-Centric</u>
- From <u>Desktop</u>   to   <u>Embedded</u> and <u>Mobile-Global</u>
- From <u>Stand-Alone</u>   to   <u>Complex Systems of Systems</u>
- <u>Long-Lived</u> and <u>Pervasive</u>
- <u>Multi-Intent</u> and <u>Multi-Faceted</u>
- Accidental complexity:
     Incompatible models, languages, platforms, methods, standards

Expanding Challenges for Software Productivity

- Larger and more diverse systems
- Pervasive demand
- Increased criticality
- Global teams and interoperability
- Workforce

Barriers and Challenges

- Intellectual control over
  unbounded complexity

- Systems are
  distributed
  heterogeneous
  hybrid
  untrusted

Representative Goal Space

- Formal, repeatable *Software Engineering* discipline
- Software that provides consistent experience
- Software that self configures and self repairs
- Systems of large numbers of low-cost, redundant throwaways
- Load-adaptive behavior
- New types of systems and system support

Sample Solution Space Needs

- Trade-offs among managed properties

- Move away from one-size-fits-all

- Multi-level middleware:
    Quality of Service (QoS) at all levels

- Support for multiple points of view

- Redefine the "what" of programming
    and the "who" of programmer

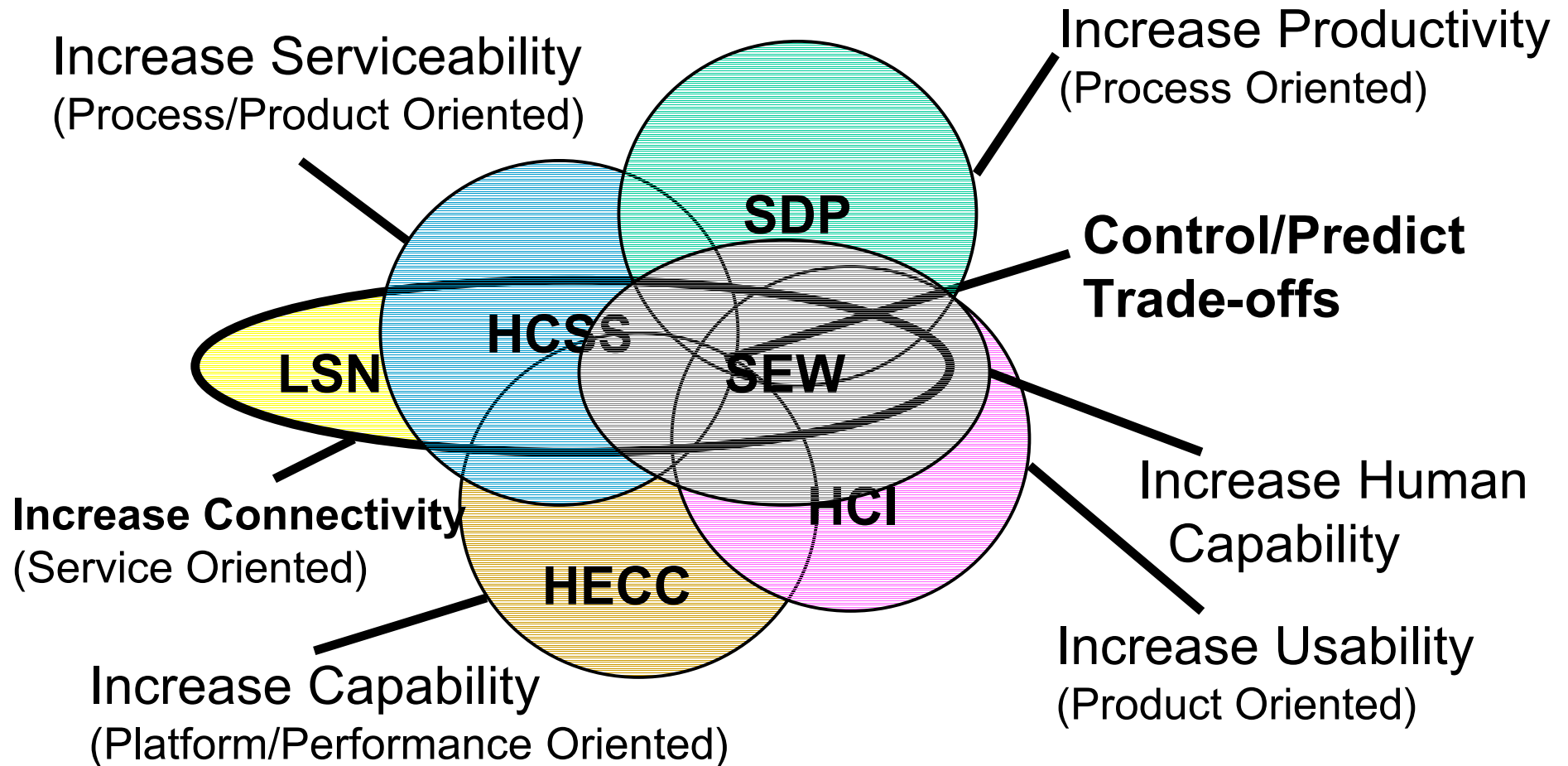- New ways to conceptualize and realize systems

Sample Solution Space Needs

- **New high-productivity construction paradigms — examples:**
    - Aspect oriented
    - Intentional approaches
    - Compositional methods
    - Multi-view abstractions and environments
    - Rich modalities for capturing programmer intentions
    - Change the meaning of "program" and "programmer"
- **New workforce solutions**
    - Education and training
    - Empowering end users: Domain Specific Languages (DSLs)
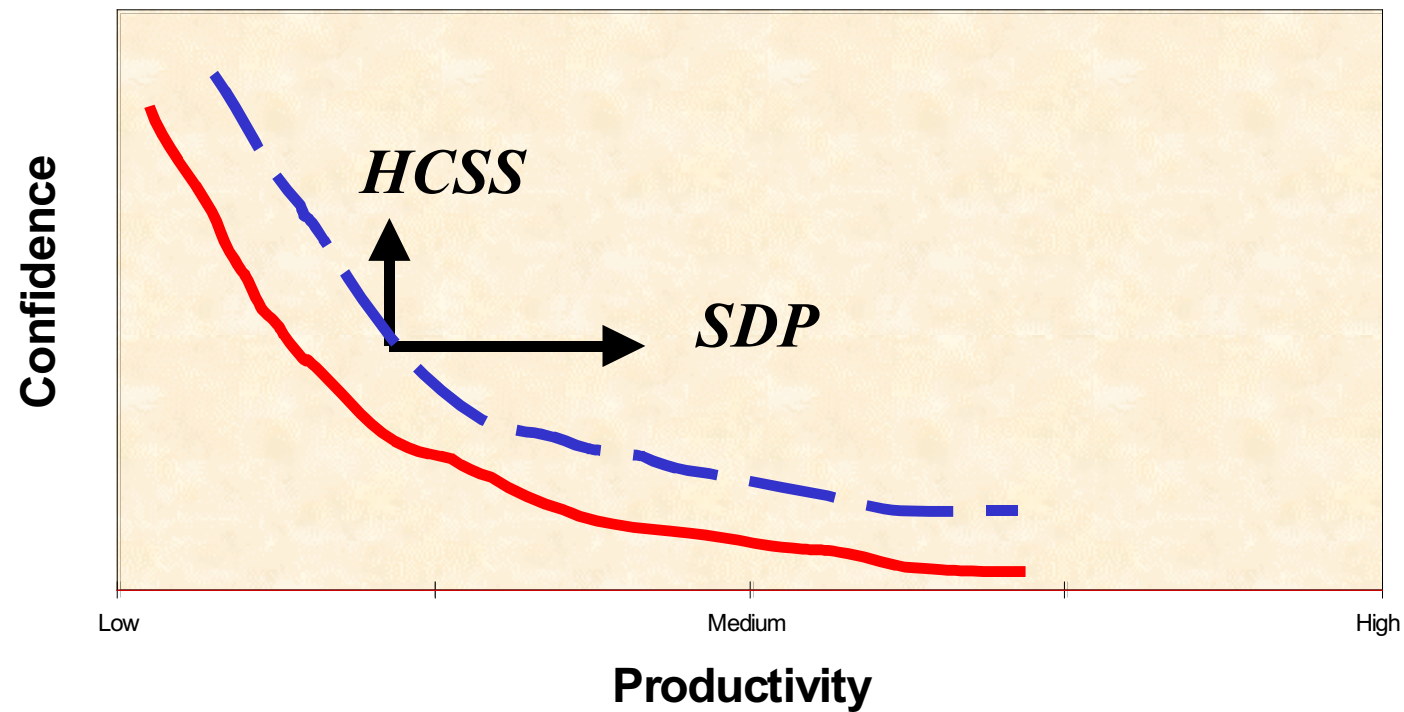   - Open software development

Related PCAs

Increase Serviceability
(Process/Product Oriented)

Increase Productivity
(Process Oriented)

SDP

Control/Predict Trade-offs

HCSS

LSN

SEW

Increase Connectivity
(Service Oriented)

HCI

Increase Human Capability

HECC

Increase Capability
(Platform/Performance Oriented)

Increase Usability
(Product Oriented)

More Productive or More Confident?

Effects of High Confidence (HCSS) and SDP Research

What Results We Need

- Input to a Research Agenda to increase software productivity
  - **Assess critical levels of research available in the community**
  - **Formulate next best investments toward goals**
  - **Produce SDP Research Needs document**
- Long-range Goals and Roadmaps
- Reasons for the government to invest in software research
- Research areas for the government to spend on
- Compelling arguments for a wide audience

RESEARCH
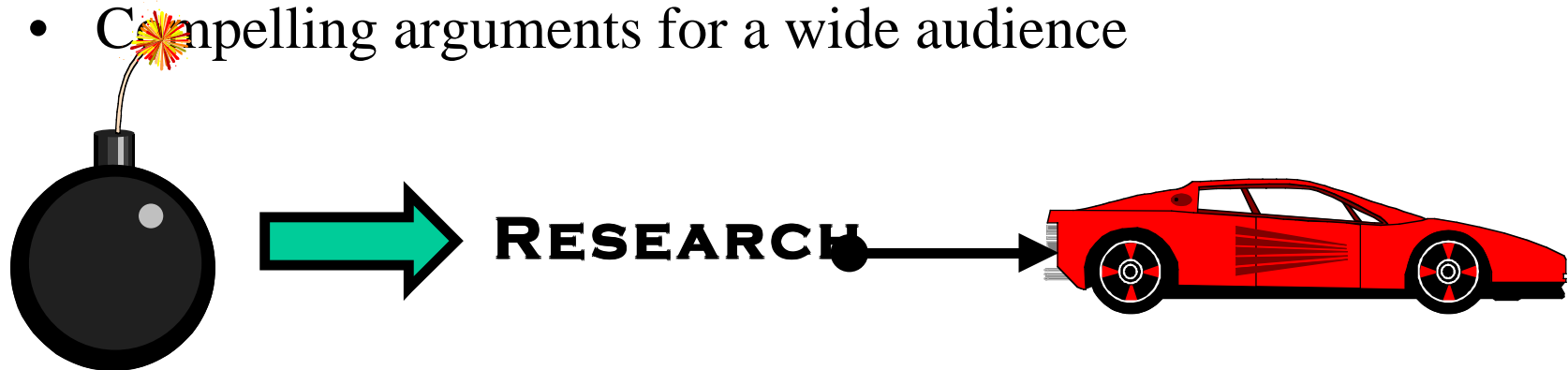
Metaphors

- Software Ecology: EcoSoft
- Software Husbandry: SoftHerding
- Software as a Global Infrastructure: SoftSource
- Software as Ubiquitous Utility: UUSoft
- Software as a Minable Resource: SoftOre
- Software as a Commodity: SoftGoods
- Software as an Engineered Artifact: Dream

End

Backup information: SDP Workshop Participants

Panel 1: Future of Software and Software Research

Bill Mark (lead)	*SRI International*
Barry Boehm	**USC**
Doris Carver	**Louisiana State University**
Bonnie John	**Carnegie Mellon University**
Benjamin Pierce	**University of Pennsylvania**
Shankar Sastry	**University of California, Berkeley**
Kevin Sullivan	**University of Virginia**

Panel 2: New Software Development Paradigms

Charles Simonyi (lead)	*Microsoft*
Grady Booch	**Rational Software**
Ralph Johnson	**University of Illinois**
Gregor Kiczales	**University of British Columbia**
John Vlissides	**IBM T.J. Watson Research**

SDP Workshop Participants

Panel 3: Software for the Real World

Don Winter (lead)	*Boeing Phantom Works*
Martin Feather	**NASA/Jet Propulsion Laboratory**
Gabor Karsai	**Vanderbilt University/ISIS**
Patrick Lardieri	**Lockheed Martin Advanced Technology**
Edward Lee	**UC Berkeley**
Cleve Moler	**MathWorks, Inc.**

Panel 4: Software for Network-Centric Distributed Systems

Rick Schantz (lead)	*BBN Technologies*
Ian Foster	**Argonne National Laboratory**
Doug Lea	**State University of New York at Oswego**
Adam Porter	**University of Maryland**
Jim Waldo	**Sun Microsystems, Inc.**

Federal government participants:

DARPA   DOE   EPA   FAA   NASA   NCO   NIST   NSF   ODUSD(S&T)   ONR   OSTP